SELF-ORIENTING WATERMARKS

Related Application Data

This application is a Continuation of 09/502,543 filed February 10, 2000 which is hereby incorporated by reference.

Technical Field

The invention relates to digital watermarking.

Background and Summary

Digital watermarking is a process for modifying a host signal or object to embed a machine-readable code into the host. The host may be modified such that the embedded code is imperceptible or nearly imperceptible to the ordinary observer upon viewing or playback, yet may be detected through an automated detection process. Most commonly, digital watermarking is applied to media such as images, audio signals, and video signals. However, it may also be applied to other types of media, including documents (e.g., through line, word or character shifting), software, multi-dimensional graphics models, and surface textures of objects.

Digital watermarking systems have two primary components: an embedding component that embeds the watermark in the host, and a reading component that detects and reads the embedded watermark. The embedding component embeds a watermark pattern by altering data samples of the host in the spatial, frequency, or other transform domains. The reading component analyzes target content to detect whether a watermark pattern is present. In applications where the watermark encodes information, the reader extracts this information from the detected watermark.

One challenge to the developers of watermark embedding and reading systems is to ensure that the watermark is detectable even if the watermarked media content is corrupted in some fashion. The watermark may be corrupted intentionally, so as to bypass its copy protection or anti-counterfeiting functions, or unintentionally through various transformations that result from routine manipulation of the content. In the case

of watermarked images, such manipulation of the image may distort the watermark pattern embedded in the image.

The invention provides a watermarking method in which attributes of the watermark used to embed information also serve to orient the watermark in the reading process. One aspect of the invention is a self orienting watermark that carries a message and has attributes that provide an orientation of the watermark signal in a host signal.

Another aspect of the invention is a method of embedding a self orienting watermark in a host signal. This method converts a message into a watermark signal having an attribute that orients the watermark in the host signal, and applies the watermark signal to the host signal. In one implementation, the method converts the message to an FSK signal. The FSK signaling frequencies have spectral attributes that orient the watermark in the host signal.

Another aspect of the invention is a method of decoding a self orienting watermark in a host signal. This decoding method uses an attribute of the watermark to determine orientation of the watermark in a host signal. The attribute provides a dual functionality of determining a watermark's orientation and carrying a message. In one implementation, this attribute is an FSK signal whose signaling frequencies help identify the orientation of the watermark and also encode a message. After finding the watermark in the host signal, the method proceeds to read the message encoded into it.

Further features and advantages of the invention will be apparent from the following detailed description and accompanying drawings.

Brief Description of the Drawings

Fig. 1A illustrates an example of a binary signal converted to a square wave. Fig. 1B illustrates an example of the binary signal converted to a continuous phase FSK signal.

Fig. 2 is a plot illustrating the FFT magnitude of the FSK signal shown in Fig. 1B.

Fig. 3 is a diagram of a watermark embedder and detector system.

Fig. 4 is a diagram illustrating an example of a watermark embedder.

Fig. 5 is a diagram illustrating another example of a watermark embedder.

Fig. 6 is a flow diagram illustrating a process for determining the orientation of a watermark signal in a signal suspected of containing a watermark.

Fig. 7 is a flow diagram illustrating a more detailed implementation of the process depicted in Fig. 6.

Fig. 8 is a flow diagram illustrating a process for extracting a message from a watermark signal embedded in another signal.

Fig. 9 illustrates an example of a computer system that serves as an operating environment for software implementations of watermarking systems described below.

Detailed Description

1.0 Overview of Watermarking Method

The following sections describe a watermarking method that converts a watermark message into a self-orienting watermark signal and embeds the watermark signal in a host signal. The spectral properties of the watermark signal facilitate its detection, even in applications where the watermarked signal is corrupted. Because of these properties, the watermark signal can perform the dual function of identifying the watermark's presence and orientation in potentially corrupted media, and also conveying a hidden message in the host signal. Such a watermark may be referred to as a self-orienting watermark.

Like conventional watermarking systems, the self-orienting watermarking systems described below have an embedder that embeds the watermark in a host signal to create a watermarked signal, and a reader that detects the watermark in a potentially corrupted watermarked signal and reads the watermark message. The embedder converts the desired message into a FSK signal. It then identifies parts of the host signal to receive the watermark and alters host signal values in those parts with corresponding values from the FSK signal. Typically, the objective of the embedder is to encode the watermark to make it imperceptible during ordinary playback of the watermarked signal, yet recoverable by the reader despite intentional or unintentional corruption.

In many applications, the host signal is multi-dimensional. For example, the samples in a still image have spatial coordinates (e.g., x and y coordinates for a 2-D image), and one or more color values depending on the color space. The samples in an audio file have a discrete time value and an intensity value. Similarly, the samples in video have spatial coordinates, temporal coordinates (e.g., the frame or field), and one or more color values. The FSK signal may be inserted along one dimension, e.g., a row of luminance values of an image, or along multiple dimensions. In addition, each dimension may encode a version of the same message, or a different message. (Although 1- and 2-D signals are most commonly watermarked, 3- and more-dimension signals may also be watermarked, e.g., wire-frame or mesh computer models of 3-D objects.)

While the following description provides specific implementation details of an image watermarking method, the general approach may be applied to other watermarkable objects (including video and audio).

2.0 FSK Signaling Background

FSK signaling is a digital communications technique in which data is conveyed by shifting between distinct frequencies of transmission. To illustrate the concept, consider the example shown in Figs. 1A-B. Suppose, for example, that one wishes to transmit some arbitrary sequence of zeros and ones. Conceptually, one can visualize the sequence as first being converted to a square wave, where the instantaneous amplitude of the square wave is represented by its corresponding binary value in the original sequence as shown in Fig. 1A. Following conversion to a square wave representation, an amplitude to frequency conversion is performed, where a lower frequency is transmitted when the square wave is in a trough and a higher frequency is sent when the square wave is cresting as shown in Fig. 1B. The result of the amplitude to frequency conversion is a typical FSK signal.

In this example, notice that when the lower frequency is transmitted the signal goes through exactly one cycle per bit. When the higher frequency is transmitted the signal goes through exactly two cycles.

This particular example is known in the literature as 2-FSK with continuous phase. See E.A. Lee, D. G. Messershmitt, *Digital Communication, Second Edition*, Chapter 6, 1994. The "2" comes from the fact that there are only two frequency states of the signal; generally M different frequencies can be used. The term "continuous phase" arises from the fact that there are no phase discontinuities between adjacent bits. There are frequency domain implications as well for continuous phase FSK as shown in Fig. 2. The peaks in the magnitude spectrum are distinct, and as such, can be used to identify the FSK signal when embedded in a host signal in watermarking applications.

3.0 A Watermark System

The following sections describe implementations of a watermark embedder, detector, and reader that operate on media signals (e.g., images, audio, etc.). The embedder encodes a message into a digital signal by modifying its sample values such that the message is imperceptible to the ordinary observer in output form. The detector captures a representation of the signal suspected of containing a watermark and then processes it to detect the watermark and determine its orientation. To extract the message, the reader uses the orientation to approximate the position of samples at encoding time and decodes the message.

Fig. 3 is a block diagram summarizing image processing operations involved in embedding and reading a watermark. There are three primary inputs to the embedding process: a host signal 100, the message 102, and a series of control parameters 104. The control parameters may include one or more keys. One key may be used to encrypt the message. Another key may be used to control the generation of a watermark carrier signal, a mapping of information bits in the message to positions in a watermark information signal, and an application of the watermark information signal to the host signal. Other parameters may include control bits added to the message.

The watermark embedding process 106 performs a watermarking function on the message to convert it to a watermark information signal. It then combines this signal with the host signal to create a watermarked signal 108.

The watermark detector 110 operates on a digitized signal suspected of containing a watermark. As depicted generally in Fig. 1, the signal may undergo various transformations 112, such as conversion to and from an analog domain, copying, editing, compression/decompression, transmission etc. Using parameters 114 from the embedder (e.g., carrier signal properties, control bits, key(s)), it performs a series of correlation or other operations on the captured signal to detect the presence of a watermark and to determine its orientation. For image signals, the orientation may be expressed in the form of orientation parameters, such as translation, rotation, scale, differential scale (shear), etc. Using these parameters to approximate the orientation of the watermark signal, the reader 116 extracts the message from the suspect signal. Some implementations do not perform correlation, but instead, use some other detection process or proceed directly to extract the watermark signal.

3.1 Embedding

Fig. 2 is a block diagram illustrating an implementation of an exemplary embedder in more detail. The embedding process begins with the message 200. As noted above, the message is binary number suitable for conversion to a watermark signal. For additional security, it may be encrypted with an encryption key 202. In addition to the information conveyed in the message, the embedder may also add control bit values to the message to assist in verifying the accuracy of a read operation. These control bits, along with the bits representing the message, are input to an error correction coding process 204 designed to increase the likelihood that the message can be recovered accurately in the reader.

There are several alternative error correction coding schemes that may be employed. Some examples include BCH, convolution coding, and turbo codes. These forms of error correction coding are sometimes used in communication applications where data is encoded in a carrier signal that transfers the encoded data from one place to another. In the digital watermarking application discussed here, the raw bit data is encoded in a fundamental carrier signal.

In addition to the error correction coding schemes mentioned above, the embedder and reader may also use a Cyclic Redundancy Check (CRC) to facilitate detection of errors in the decoded message data.

The error correction coding function 204 produces a string of bits, termed raw bits 206, that are embedded into a watermark information signal. Using a carrier signal 208 and an assignment map 210, the illustrated embedder encodes the raw bits in a watermark information signal 212, 214. The carrier signal may be a random or pseudo-random signal. The assignment map specifies where to place the watermark information in the host signal. In some applications, the embedder may encode a different message in different locations of the signal.

While the precise functions and processing order may vary with the implementation, the embedding process proceeds generally as follows. The embedder modulates the carrier signal with the raw bit signal. It then FSK modulates the resulting carrier signal. Finally, it maps the FSK signal to a location or locations in the host signal. This process creates a watermark information signal. The processing order may be rearranged, and some processing stages may be omitted in some implementations. For instance, some implementations may apply FSK modulation to the raw bit signal without modulating the raw bits with a carrier signal. The carrier signal may be used to spread a raw bit over a pseudo-random signal. While advantageous in some applications, this spreading operation is not necessary.

Having summarized a general embedding framework, a number of points can be made. First, the embedder may perform a similar approach in any transform domain. For example, the FSK modulated signal may be mapped to samples in the spatial or temporal domain or some other transform domain.

Second, the specific mathematical relationship among the raw bits, the carrier, the FSK modulator, and the assignment map may vary with the implementation. For example, the message may be convolved with the carrier, multiplied with the carrier, added to the carrier, etc.

Third, the carrier signal may remain constant for a particular application, or it may vary from one message to another. For example, a secret key may be used to generate the carrier signal.

Fourth, the assignment map may map a raw bit (or its corresponding modulated signal) to a single location or many locations or orientations, in one or more transform domains. For example, the FSK signal could be applied to the rows and columns of image samples in the spatial domain.

Fifth, the assignment map may remain constant, or it may vary from one message to another. In addition, the carrier signal and map may vary depending on properties of the host signal. In sum, there are many possible design choices within the implementation framework described above.

Returning to Fig. 2, the embedder makes a perceptual analysis 218 of the host signal 220 to identify portions of the signal that can withstand more watermark signal content without substantially impacting fidelity. For images, the perceptual analysis identifies portions where there is more image activity. In these areas, the sample values are changing more than other areas and have more signal strength. The output of the perceptual analysis is a perceptual mask 222 that represents signal activity. For example, the mask may be implemented as an array of multipliers, which selectively increase the signal strength of the watermark information signals in areas of greater signal activity. The embedder combines (224) the watermark information signal and the perceptual mask to yield the watermark signal 226. Finally, it combines (228) the host signal 220 and the watermark signal 226 to create the watermarked signal 230.

In one implementation where the watermark signal is inserted in the spatial domain of an image, the embedder adds the image samples in the watermark signal to the corresponding samples in the input image to create the watermarked image 230. In other implementations, the embedder may perform alternative functions to combine the watermark signal and the input image (e.g., multiplication). The net effect is that some image samples in the input image are adjusted upward, while others are adjusted downward. The extent of the adjustment is greater in areas of the image having greater signal activity.

3.2 Example Embedder Implementation

The following sections describe an implementation of the digital image watermark embedder depicted in Fig. 5. The embedder inserts a self-orienting watermark into the spatial domain of the host image. The watermark serves a dual function of conveying a message and identifying the watermark location in the image.

The embedder inserts watermark components in rows and columns of luminance samples of the host image at a pre-determined resolution (e.g., 100 dpi - 300 dpi). The message payload size varies from one application to the next. Typically, the payload ranges from 2-128 bits.

3.2.1 Encoding the Message

The embedder converts binary message bits into a series of binary raw bits that it hides in the host image. As part of this process, a message encoder 300 appends certain known bits to the message bits 802. It performs an error detection process (e.g., parity, Cyclic Redundancy Check (CRC), etc.) to generate error detection bits and adds the error detection bits to the message. An error correction coding operation then generates raw bits from the combined known and message bit string.

For the error correction operation, the embedder employs convolution coding. Other alternatives include BCH and turbo coding.

3.2.2 Spread Spectrum Modulation

The embedder uses spread spectrum modulation as part of the process of creating a watermark signal from the raw bits. A spread spectrum modulator 304 spreads each raw bit into a number of "chips." The embedder generates a pseudo random number that acts as the carrier signal of the message. To spread each raw bit, the modulator performs an exclusive OR (XOR) operation between the raw bit and each bit of a pseudo random binary number of a pre-determined length. Preferably, the pseudo random number should contain roughly the same number of zeros and ones. The spread spectrum modulator

produces a binary sequence having a set of binary numbers corresponding to each raw bit.

3.2.3 FSK Modulation

Next, an FSK modulator 305 converts the spread spectrum signal into an FSK signal. In particular, the FSK modulator uses 2-FSK with continuous phase: a first frequency represents a zero; and a second frequency represents a one. The FSK modulated signal is applied to rows and columns of the host image. Each binary value in the input signal corresponds to a contiguous string of at least two samples in a row or column of the host image. Each of the two frequencies, therefore, is at most half the sampling rate of the image. For example, the higher frequency may be set at half the sampling rate, and the lower frequency may be half the higher frequency.

3.2.4 Mapping the Watermark Information Signal

The embedder maps the FSK modulated signal to locations in the host image. An assignment map 306 assigns the FSK modulated signal, or segments of it, to selected rows and columns of the host image. Depending on the message and image size, the embedder may map the same message to each row and column. In some applications, the embedder may map parts of a message, or different messages, to different rows or columns of the image. Also, it is possible to map the FSK signal in different directions other than the vertical and horizontal directions.

3.2.5 Computing the Watermark Information Signal

The embedder applies the watermark information signal to the host image by adding samples of the watermark to corresponding samples in the host image. Before adding the watermark information signal to the host, it converts the watermark information signals to delta values that alter corresponding samples in the host image so as to embed the watermark information (330).

At this stage, the embedder has produced an FSK modulated signal and mapped it into a two dimensional image space. Next, it assigns a delta to each sample of the

watermark based on the value of the FSK signal (or signals) mapped to that location. In particular, for a mapped sample having a value of one, it adds to the corresponding luminance value, and for a mapped sample having a value of zero, it subtracts from the corresponding luminance value.

In an alternative implementation, the embedder may apply the watermark signal by multiplying it with the host image.

Optionally, a gain controller may then adjust the magnitude of each sample in the watermark information signal.

3.2.6 Gain Control and Perceptual Analysis

Though not necessary, it is often useful to control the gain of the watermark information signal. Through gain control, the embedder may increase the strength of the signal while reducing its perceptibility to ordinary observers. The embedder shown in Fig. 5 provides an example of a gain controller that takes into account the image and the watermark information signal to calculate an array of gain multipliers to be applied to the watermark information signal.

Fig. 5 depicts the gain controller used in the embedder. Note that the gain controller operates on the luminance data 308, the watermark signal, and a global gain input 310, which may be specified by the user. A perceptual analyzer component (312) of the gain controller performs a perceptual analysis on the luminance samples to identify areas that can tolerate a stronger watermark signal without substantially impacting visibility. In places where the naked eye is less likely to notice the watermark, the perceptual analyzer increases the strength of the watermark. Conversely, it decreases the watermark strength where the eye is more likely to notice the watermark.

The perceptual analyzer shown in Fig. 5 performs a series of filtering operations on the image block to compute an array of gain values. There are a variety of filters suitable for this task. These filters include an edge detector filter that identifies edges of objects in the image, a non-linear filter to map gain values into a desired range, and averaging or median filters to smooth the gain values. Each of these filters may be implemented as a series of one-dimensional filters (one operating on rows and the other

on columns) or two-dimensional filters. The size of the filters (i.e. the number of samples processed to compute a value for a given location) may vary (e.g., 3 by 3, 5 by 5, etc.). The shape of the filters may vary as well (e.g., cross-shaped, square etc.). The perceptual analyzer process produces a detailed gain multiplier. The multiplier is a vector with elements corresponding to image samples.

The embedder may optionally compute another type of gain, called asymmetric gain (318). Asymmetric gain helps to increase the chances of an accurate read of the watermark message. This component of the gain controller analyzes the filtered luminance samples to determine whether they have values that are consistent with the watermark information signal. To illustrate the concept, consider a segment of the FSK signal representing a value of one. In the watermark information signal, this segment may correspond to a row of four luminance samples forming a wave with energy concentrated at the FSK frequency that represents a one. If the corresponding samples in the host signal already oscillate in a similar fashion, then the asymmetric gain may leave the watermark information signal unchanged. Conversely, if the corresponding samples do not oscillate in the same fashion, the asymmetric gain for selected samples may be set so that the resulting watermarked samples are more likely to produce the desired value in a read operation.

In summary, the gain controller shown in Fig. 5 has three sources of gain: the detailed gain from a perceptual analysis of the host image, the global gain provided as input, and the asymmetric gain from a comparison of the host image to the watermark information signal. The embedder multiplies the individual gain components together to get the composite gain.

3.2.7 Forming the Watermarked Signal

To compute the watermark signal, the embedder multiplies the delta values produced in block 330 by the composite gain. It then adds the samples of the watermark signal to corresponding samples of the host signal to produce the watermarked image.

3.3 Detecting an FSK Watermark

3.3.1 Establishing Orientation

Fig. 6 is a flow diagram illustrating a process for detecting a watermark and determining its orientation in a signal suspected of containing a self-orienting watermark. First, the detector transforms the image data 400 to another domain 402, (e.g., a spatial frequency domain, and then performs a series of correlation or other detection operations 404. The correlation operations match the watermark pattern with the target image data to detect the presence of the watermark and its orientation parameters 406.

Fig. 7 is a flow diagram illustrating a detector implementation in more detail. The detector transforms the image samples (410) to the color space in which the watermark was embedded (e.g., into luminance samples) (412). Optionally, it may perform various filtering operations to enhance the detection process. For example, one form of enhancement is to identify and remove signal content that would otherwise tend to obscure the watermark pattern and make it more difficult to compute its orientation. Since this detector determines the watermark's rotation angle and scale based on an FSK signal at known frequency ranges, it may improve the correlation process by removing signal content at other frequency ranges. As such, a filter may be used to highlight edges in the signal and then reduce or eliminate them (414). One type of filter suitable for this task is a multi-axis high pass filter (e.g., LaPlacian, Sobol, etc.).

Next, it prepares the image signal for a Fast Fourier Transform (FFT) by applying a window function (416). It then performs an FFT on the luminance signal to produce a Fourier magnitude signal (418). Optionally, it may filter the FFT magnitude array to boost peaks and suppress the fall-off region around these peaks.

To determine rotation and scale parameters of the watermark's orientation, the detector maps the signal to a log-polar coordinate space (422) and correlates the mapped signal with a detection pattern (424). One axis in the log-polar space represents a log of the scale parameter, and the other represents the rotation angle parameter. This mapping process is sometimes referred to as a Fourier Mellin transform. The detector correlates the resulting sample array in the log polar coordinate system with a log polar representation of a detection pattern. The detection pattern represents cosine waves at the

FSK signaling frequencies. In the spatial frequency domain (e.g., FFT magnitude plot of the detection pattern), the detection pattern has energy focused at grid points located at the signaling frequencies. (This arrangement is further detailed in patent 5,862,260, and in application 09/452,023, the complete disclosures of which are incorporated herein by reference.)

When FSK signaling is applied to the rows and columns, the FFT magnitude of pure cosine waves at the signaling frequencies produces grid points along the vertical and horizontal axes in a two-dimensional frequency spectrum. If different signaling frequencies are used for the rows and columns, these grid points will fall at different distances from the origin. These grid points, therefore, may form a detection pattern that helps identify the rotation angle of the watermark in a suspect signal. Also, if an image has been rotated or scaled, the FFT of this image will have a different frequency spectrum than the original image. In particular, the peaks once located at the FSK signaling frequencies will be rotated and scaled. By aligning the peaks in the distorted image with the original peaks, the detector can re-align the image to its original orientation.

A variety of correlation methods may be used to find the rotation and scale parameters that optimize the correlation between the suspect signal and the detection pattern. This detector implementation uses a form of a generalized matching filter (GMF) (424). The GMF performs yet another FFT on the suspect signal and the pattern, multiplies the resulting signals (i.e. computes the dot product of the two FFT arrays), and then computes the inverse FFT of the product. The product comprises an array of correlation values in log-polar space. The detector may use interpolation to find the inter-sample location of the rotation and scale vector that provides the best correlation. It may use the same approach to find a set of rotation-scale vectors with the top correlation values (426).

At this stage, the detector has recovered rotation and scale parameters that estimate the orientation of the watermark in the suspect signal. Before proceeding, it is worth noting that there are several alternative approaches to estimating the rotation and scale parameters. Recall that the FFT magnitude of the FSK signal has peaks at the signaling frequencies as shown in Fig. 2. One way to determine the rotation is to step

through a set of candidate rotation angles, apply the candidates to the image, and find the rotation angle that yields an FFT magnitude with the highest peak at the signaling frequencies. Since the image is two dimensional, the embedder can insert the FSK signal in the rows and columns of the image. Thus, this process may be applied separately to the rows and columns to find rotation angles that maximize the peaks in each dimension. Alternatively, a similar process may be applied to both rows and columns simultaneously. In this case, the approach is similar, except that the FFT magnitude is two-dimensional, and there are four peaks (two for the rows and two for the columns).

Once rotation is established, the detector derives the scale parameter from the location of the peaks. If the resolution of the image has been increased relative to its resolution at embedding, then the location of the peaks will shift toward the origin in the Fourier spectrum. Conversely, the location of the peaks will shift away from the origin if the resolution has decreased. If the watermark is embedded in both the rows and columns, a separate scale parameter may be established for each dimension.

As another alternative, the detector may employ projection techniques to ascertain orientation. In one such approach, the detector takes the absolute value of the FFT of each row of the watermarked image, and accumulates the result for all rows. If the watermarked image is correctly aligned, this process will produce two peaks exactly at the FSK signaling frequencies. If there is scaling, the location of the peaks changes. If the image is rotated, the result will appear like scaling combined with a loss in peak strength. The amount of rotation and scale can be derived from the position and amplitude of the peaks.

If the message content within the watermark is unknown, finding the origin of the watermark in the suspect signal presents a challenge. Generally, the implementer of the system can address this challenge by giving the watermark an attribute that enables the detector to derive it, even if the signal has been corrupted in some fashion. Just as the FSK frequencies represent an attribute that allows the detector to recover rotation and scale, the embedder may give the watermark another attribute that allows the detector to recover translation parameters that specify an origin or point of reference from which to

orient the reader. After correcting for rotation and scale, the detector can then search for the attribute for recovering the translation parameters.

One such attribute is a sequence of phase offsets. Notice that in Fig. 1B, the value of the FSK signal at a bit transition is always "1." Measured from bit to bit, the signal contains only integer cycles of a pure cosine, i.e. no phase offset. Instead of embedding with the same phase offset every row, the embedder may insert an offset that is row dependent. Knowing how the embedded phase offset varies with each row, the detector can establish a row of origination. The same procedure could be used to find the column of origination. The intersection of the two then defines the origin.

Another attribute is a sequence of FSK signaling frequencies. For example, the embedder may vary the FSK signaling frequencies over a sequence of two or more rows and columns. The detector can then identify these frequencies by computing the FFT magnitude of the rows and columns and identifying the peaks. Knowing the sequence of signaling frequencies, the detector can establish a row of origination. The same procedure can be used to find the column of origination. Again, the intersection of the row and column of origination defines the origin. Note that the detector may perform the process of identifying these peaks as part of the process of determining the rotation angle.

Note that if the embedder inserts an FSK watermark in each row (or column) at the same phase offset and signaling frequencies, artifacts will be noticeable. Thus, it is advantageous to vary the phase offset and/or signaling frequencies across rows (or columns) to improve image fidelity. Also, it is advantageous to use different signaling frequencies in the rows and columns.

Yet another attribute is a sequence of known bits appended to the watermark. The detector may then search for these known bits, after adjusting for rotation and scale. To search, the detector invokes a reader to extract a message from the rows and columns. It then looks for the known bits to identify the row and column of origination.

3.4 Reading the FSK Watermark

If a watermark is detected, the reader extracts the watermark information signal from the image data (optionally by first re-orienting the data based on the orientation

parameters). Fig. 5 is flow diagram illustrating a process of extracting a message from re-oriented image data 500.

Once orientation is established, the detector invokes the reader on the rows and columns. For the rows, the reader scans the input image row of interest into two independent channels (502), each of which is band-pass filtered at one of the two possible FSK frequencies (504). Then, for a given bit location, the reader decides that the bit is a zero if there is more energy in the output channel of the lower frequency filter (506). The reader decides that the bit is a one if there is more energy in the higher frequency filter (506). An exemplary implementation embeds on the order of four pixels per bit at 100dpi, or 40 bits per inch. In general, with more pixels per bit, the individual decisions on the bits will be more reliable. As is known from the assignee's prior art, redundant signaling would be used here as well.

The reader converts the bit value to a corresponding raw bit value and buffers the value for the associated raw bit (508). In particular, if the message has been spread spectrum modulated, then the reader performs an inverse of the modulation operation to recover a candidate value for a corresponding raw bit and buffers the candidate value. In this reading process, the reader uses the assignment map to map the extracted candidate value to its corresponding raw bit position.

Next, the reader compiles the candidate values for each raw bit to compute a final value for each raw bit position (510). It compiles the candidates for the raw bits extracted from each of the tows. For a given raw bit, if more candidates are a one, then the final value is set to a one (and vice versa if more candidates are a zero). Finally, it performs error correction decoding to reconstruct the original message (512).

To extend to reading in the other dimension, one may use a similar approach. One could either embed the same bit at identical locations in both directions, or embed directionally dependent information. In the former case, the reader can achieve a more accurate message recovery by combining candidates from both rows and columns. In the latter, potentially more information would be embedded.

<u>4.0 Embedding Multiple Watermarks</u>

The self-orienting watermark described above may be embedded along with other watermarks into a host signal. For example, an FSK watermark may be combined with a spatial domain watermark, or other type of watermark inserted in another transform domain (e.g., wavelet, Discrete Cosine Transform, Discrete Fourier Transform domains). In such multi-watermark schemes, the self-orienting watermark may be used to establish orientation of other watermarks in the host signal. Before reading another watermark, the detector can mitigate the impact of interference due to the self-orienting watermark. For an FSK watermark, for example, the detector could apply a band-stop filter at the FSK signaling frequencies. This filtering operation would mitigate the interference due to the FSK signal when attempting to read other watermark messages.

5.0 Applications

There are many applications for self-orienting watermarks. The watermarks may carry data, machine instructions, and/or links to other data or instructions. The link may be implemented as an address or reference to a database or other resource located on the same device as the detector or on a remote device (e.g., a computer on the Internet). Whether stored within the message, or linked by the message, the data and/or machine instructions may be used to authenticate the host signal, monitor the host signal (e.g., broadcast monitoring), identify its owner, control copying of the host signal, provide additional information related to the host signal, etc.

6.0 Operating Environment for Computer Implementations

Figure 9 illustrates an example of a computer system that serves as an operating environment for software implementations of the watermarking systems described above. The embedder and detector implementations are implemented in C/C++ and are portable to many different computer systems. Fig. 9 generally depicts one such system.

The computer system shown in Fig. 9 includes a computer 1220, including a processing unit 1221, a system memory 1222, and a system bus 1223 that interconnects various system components including the system memory to the processing unit 1221.

The system bus may comprise any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using a bus architecture such as PCI, VESA, Microchannel (MCA), ISA and EISA, to name a few.

The system memory includes read only memory (ROM) 1224 and random access memory (RAM) 1225. A basic input/output system 1226 (BIOS), containing the basic routines that help to transfer information between elements within the computer 1220, such as during start-up, is stored in ROM 1224.

The computer 1220 further includes a hard disk drive 1227, a magnetic disk drive 1228, e.g., to read from or write to a removable disk 1229, and an optical disk drive 1230, e.g., for reading a CD-ROM or DVD disk 1231 or to read from or write to other optical media. The hard disk drive 1227, magnetic disk drive 1228, and optical disk drive 1230 are connected to the system bus 1223 by a hard disk drive interface 1232, a magnetic disk drive interface 1233, and an optical drive interface 1234, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions (program code such as dynamic link libraries, and executable files), etc. for the computer 1220.

Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and an optical disk, it can also include other types of media that are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, and the like.

A number of program modules may be stored in the drives and RAM 1225, including an operating system 1235, one or more application programs 1236, other program modules 1237, and program data 1238.

A user may enter commands and information into the personal computer 1220 through a keyboard 1240 and pointing device, such as a mouse 1242. Other input devices may include a microphone, sound card, radio or television tuner, joystick, game pad, satellite dish, digital camera, scanner, or the like. A digital camera or scanner 43 may be used to capture the target image for the detection process described above. The camera and scanner are each connected to the computer via a standard interface 44. Currently, there are digital cameras designed to interface with a Universal Serial Bus (USB),

Peripheral Component Interconnect (PCI), and parallel port interface. Two emerging standard peripheral interfaces for cameras include USB2 and 1394 (also known as firewire and iLink).

In addition to a camera or scanner, watermarked images or video may be provided from other sources, such as a packaged media devices (e.g., CD, DVD, flash memory, etc), streaming media from a network connection, television tuner, etc. Similarly, watermarked audio may be provided from packaged devices, streaming media, radio tuner, etc.

These and other input devices are often connected to the processing unit 1221 through a port interface 1246 that is coupled to the system bus, either directly or indirectly. Examples of such interfaces include a serial port, parallel port, game port or universal serial bus (USB).

A monitor 1247 or other type of display device is also connected to the system bus 1223 via an interface, such as a video adapter 1248. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The computer 1220 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 1249. The remote computer 1249 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 1220, although only a memory storage device 1250 has been illustrated in Figure 9. The logical connections depicted in Figure 9 include a local area network (LAN) 1251 and a wide area network (WAN) 1252. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 1220 is connected to the local network 1251 through a network interface or adapter 1253. When used in a WAN networking environment, the personal computer 1220 typically includes a modem 1254 or other means for establishing communications over the wide area network 1252, such as the Internet. The modem 1254, which may be internal or external, is connected to the system bus 1223 via the serial port interface 1246.

In a networked environment, program modules depicted relative to the personal computer 1220, or portions of them, may be stored in the remote memory storage device. The processes detailed above can be implemented in a distributed fashion, and as parallel processes. It will be appreciated that the network connections shown are exemplary and that other means of establishing a communications link between the computers may be used.

7.0 Concluding Remarks

The watermarking technology detailed herein can be employed in numerous diverse applications. See, e.g., the applications for watermarking detailed in commonly-owned patent 5,862,260, and copending applications 09/292,569, 60/134,782, 09/343,104, 09/473,396, 09/476,686, and 60/141,763.

Having described and illustrated the principles of the invention with reference to a specific embodiment, it will be recognized that the principles thereof can be implemented in other, different, forms.

The particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.

In view of the wide variety of embodiments to which the principles of the invention can be applied, it should be recognized that the detailed embodiment is illustrative only and should not be taken as limiting the scope of the invention. Rather, I claim as my invention all such embodiments as may come within the scope and spirit of the following claims, and equivalents thereto.